Scalable High-Performance Heuristics for Sensor Placement in Water Distribution Networks*

Jean-Paul Watson[†] William E. Hart [†] Jonathan W. Berry [†]

Abstract

A number of algorithms have been developed to solve the problem of where to place a limited number of sensors in a water distribution network such that public health protection from accidental or intentional contaminant injections is maximized. However, the ability of these algorithms to solve real-world, large-scale sensor placement problems has yet to be demonstrated. Existing research exhibits at least one of three fundamental flaws. First, most algorithms are tested exclusively on small-scale networks, leaving open the question of scalability. Second, many algorithms are heuristic in nature and no effort has been made to establish empirical or theoretical performance bounds. Third, the modeling assumptions underlying some algorithms are physically unrealistic, raising questions regarding the utility of the resulting solutions in operational settings. We describe a modeling methodology that precisely captures the impact of contaminant injection on a distribution network. Using exact methods, we generate provably optimal sensor placements for networks containing up to roughly 3,000 junctions using high-performance computing platforms; the magnitude of the model currently prevents solution for larger networks. Next, we use a simple heuristic based on GRASP, local search, and path relinking to quickly generate solutions to even larger networks containing up to 12,000 junctions. Where solvable via exact methods, we demonstrate that the heuristic yields solutions that are globally optimal. These results conclusively demonstrate the practical application of this heuristic to solve very large sensor placement problems under realistic modeling assumptions, and uniquely provides an empirical performance bound for the algorithm.

1 Introduction

Research on the problem of where to place sensors in water distribution networks to minimize the damage incurred by the intentional injection of chemical and biological

*Sandia is a multipurpose laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000. The authors would also like to thank Jim Uber for useful feedback during various discussions relating to this research.

†Discrete Algorithms and Mathematics Department, Sandia National Laboratories, P.O. Box 5800, MS 1110, Albuquerque, NM 87185-1110 USA, {jwatson,wehart,jberry}@sandia.gov.

contaminants has intensified since the attacks of 9/11; the vulnerability of these systems has become more widely appreciated. Accurate models of the sensor placement optimization problem now exist, in addition to methodologies for generating the associated model parameters. However, while algorithms for generating solutions to these models have been introduced, their applicability to large-scale, real-world water distribution networks is far from clear. First, most algorithms are tested exclusively on small-scale networks, leaving open the question of scalability. Second, many algorithms are heuristic in nature and no effort has been made to establish empirical or theoretical performance bounds. Third, the modeling assumptions underlying some algorithms are physically unrealistic, raising questions regarding the utility of the resulting solutions in operational settings.

In this paper, we analyze the scalability of exact and heuristic algorithms for sensor placement optimization under very realistic modeling assumptions. We show that exact integer programming methods can generate optimal solutions to moderately-sized networks, while heuristic methods can quickly locate optimal solutions to these same networks, but are further capable of generating (possibly sub-optimal) solutions to very large networks containing up to 12,000 junctions. This is the first demonstration of truly scalable algorithm performance for heuristic sensor placement optimization.

The rest of this paper is organized as follows. We briefly summarize and categorize prior research on sensor placement optimization in §2. In §3, we describe our formulation of the sensor placement problem and relate it to the well-known p-median problem. Both exact and heuristic solution approaches are introduced in §4, while the empirical performance of these methods is analyzed in §5. We discuss the implications of our results in §6 and recap our primary conclusions in §7.

2 Optimization for Sensor Placement: Background

Conceptually, the objective in a sensor placement optimization problem (SPOP) is simple: to place a limited number of sensors in a water distribution network such that the impact to public health due to the accidental or intentional injection of contaminant is minimized. The broader research community has yet to arrive at a more specific, concrete definition that is widely (or even narrowly) agreed upon; research typically differs in terms of the precise definition of public health impact, the assumed characteristics of the deployed sensors, the fidelity of the contaminant transport simulation, and a host of other details. However, existing formulations of the SPOP can be usefully delineated in terms of the fidelity with which water quality changes resulting from an injection is captured. Two broad categories in the current literature can be identified, which we refer to simply as static and dynamic.

In a *static* formulation of the SPOP, the impact of an attack at a particular network junction is <u>estimated</u> by analyzing some combination of (1) flow directions and velocities obtained via hydraulic simulation, (2) pipe lengths, and (3) junction demands. A prominent example of a static SPOP formulation is described by Kessler et al. [1998], and is based on the notion of an auxiliary network. An auxiliary network is a directed graph $G = (V, E)$ where elements of the set V represent nodes, e.g., junctions and tanks, of the distribution network. The edge set E is determined via analysis of hydraulic simulation outputs. For each pair of nodes v_i and v_j for which there is flow from v_i to v_j at any point in the simulation, a directed edge $e = (v_i, v_j)$ is added to E. Edges in $e = (v_i, v_j) \in E$ are weighted by the average velocity from v_i to v_j over the course of the simulation. The auxiliary graph is used in conjunction with the network pipe lengths to compute the shortest travel time between all pairs of vertices in the network. The travel times are then used to estimate the network-wide impact of an attack α at v_i if α is first detected by a sensor located at a vertex v_j; the specific measure of health impact considered by Kessler et al. is the total volume of contaminated water consumed before detection by at least one sensor. Kessler et al. solve their static formulation of the SPOP via heuristic solution of a corresponding set cover problem.

Berry et al. [2003] introduce a static SPOP formulation in which the objective is to minimize the expected fraction of the population exposed to an injected contaminant. Here, hydraulic simulation results are used to compute a fixed flow orientation for each pipe in the network over a series of p distinct non-overlapping time intervals, referred to as patterns. The formulation is time-independent, in that travel times are not considered; rather, a node v_j is protected against an attack at vertex v_i if and only if there is a sensor capable of detecting the flow between v_i and v_j. Watson et al. [2004] generalize the Berry et al. formulation to consider a range of optimization objectives, some of which account for travel times in a manner consistent with that of the Kessler et al. formulation. Both Berry et al. and Watson et al. solve the resulting SPOP formulations via exact solution of corresponding mixed-integer programs.

There are two key assumptions underlying any static SPOP formulation, e.g., that of Kessler et al. First, factors such as contaminant dilution, concentration level, and mode of attack are not modeled. Rather, the static SPOP simply tracks the *projected* presence or absence of contaminant at various network points over time, and assumes identical contaminant and water flow dynamics. Second, the contaminant transport model is based on aggregated flow velocities, such that the true dynamics of the underlying flow are only approximated. Each of these assumptions represents a potentially significant deviation from reality, and the impact of these approximations on the quality of solutions to the SPOP are currently poorly understood.

In contrast, *dynamic* SPOP formulations correct for each of the aforementioned deficiencies by precisely characterizing the impact of an attack at a given network junction on the rest of the network. First introduced by Ostfeld and Salomons [2004], dynamic SPOP formulations use detailed water quality simulation results to compute contaminant concentration time-series for each junction in the network. These time-series can be used to determine the impact of an attack α at v_i if α is first detected by a sensor located at a vertex v_j. In addition to accuracy improvements relative to their static counterparts, dynamic SPOP formulations have the added advantage that a full range of attack types and sensor characteristics can be modeled, as the network response is completely specified by contaminant level time-series at each network junction; the accuracy of the formulation is strictly limited by the accuracy of the water quality simulation. Mirroring the earlier approach of Kessler et al., Ostfeld and Salomons solve their dynamic SPOP via solution of a corresponding set covering problem. However, the optimization objective is more realistic: to minimize the volume of polluted water possessing a concentration of pollutant higher than a minimum hazard level. Most recently, Berry et al. [2004] discuss a dynamic SPOP formulation for minimizing the expected volume of contaminated water consumed before detection, which is expressed and solved as a mixed-integer program.

Finally, we observe that the accuracy of dynamic SPOP formulations comes with a price, specifically in the form of a very large number of computationally intensive water quality simulations; in contrast, static SPOP formulations are based strictly on comparatively cheap hydraulic simulations. We explore this issue further in both §5 and §6.

3 Sensor Placement and the p-Median Problem

We now introduce the specific formulation of the dynamic SPOP used in our analysis. Our objective is to minimize the total volume of contaminated water consumed, at any concentration level. We assume that we have a budget of p of sensors that can be placed at any junction in a distribution network, each sensor is capable of detecting any concentration level of contaminant, and a general alarm is immediately raised when contaminant is detected such that all further consumption is prevented. As discussed in §2, we observe that none of these assumptions are binding, and can be relaxed without impacting the mathematical structure of our formulation. We view the structure of a water distribution network as an undirected graph $S = (V, E)$; elements of the set V represent junctions and sources, while elements of the set E represent pipes, pumps, and valves.

Let A denote the set of attacks against which a sensor configuration consisting of p sensors is intended to protect. We assume attacks can occur at any vertex $v \in V$ of the network, i.e, injection via backflow is possible. Elements of $a \in A$ are quadruples of the form $a = (v_x, t_s, t_f, X)$, where $v_x \in V$ is the attack vertex, t_s and t_f are the attack start and stop times, and X is the attack profile (e.g., arsenic injected at a particular concentration at a given rate). For each attack $a \in A$, we use existing water quality analysis software (e.g., as found in EPANET [Rossman, 1999]) to compute the contaminant concentration at each node in the network from time t_s to an arbitrary point $t_h \geq t_f$ in the future. The results of such an analysis are expressed in terms of concentration time-series τ_j for each $v_j \in V$, with samples at regular (arbitrarily small) intervals within $[t_s, t_h]$. Using the set of τ_j in conjunction with demand profiles, it is straightforward to compute the total volume of contaminated water $d_a(t)$ consumed (network-wide) due to an attack a at any given point at time $t \in [t_s, t_h]$. Next, let γ_{aj} denote the earliest time t at which a hypothetical sensor at vertex v_j can detect contaminant due to an attack a; $\gamma_{aj} = t_h$ if no contaminant ever reaches v_j, and γ_{aj} can be easily computed from τ_j. Finally, we define $d_{aj} = d_i(\gamma_{aj})$, i.e., the total volume of contaminant consumed due to an attack a if the attack is first detected by a sensor at v_j.

Given a set A of attack scenarios, a set V of network vertices, and a set d_{aj} of impact parameters, we take as our design objective the minimization of the aggregate impact I

over all attack scenarios, where

$$I = \sum_{i=a}^{|A|} \sum_{j=1}^{|V|} d_{aj} x_{aj} \tag{1}$$

subject to the constraints

$$\sum_{j=1}^{|V|} x_{aj} = 1, \forall a \in A \tag{2}$$

$$x_{aj} \leq s_j, \forall a \in A, \forall j \in V \tag{3}$$

$$\sum_{j=1}^{|V|} s_j = p \tag{4}$$

$$0 \leq x_{aj} \leq 1, s_j \in \{0, 1\}, \forall a \in A, \forall j \in V. \tag{5}$$

A variable s_j (Constraint 5) indicates whether one of the p available sensors is placed on vertex v_j, while Constraint 4 requires that a total of exactly p sensors be placed. A variable x_{aj} (Constraint 5) indicates whether an attack $a \in A$ is detected by a sensor at vertex v_j; Constraint 3 enforces the condition that detection can only occur at v_j if a sensor is placed there. Finally, Constraint 2 requires that detection of each attack $a \in A$ be assigned to a single vertex v_j; in other words, there is always a first vertex in the network to detect an attack. We observe that this formulation is conceptually identical to the dynamic SPOP introduced by Berry et al. [2004]. Our variant is more explicit, for reasons discussed below, in that d_{aj} are defined for all possible combinations of attack $a \in A$ and vertex $v_j \in V$ – despite the fact that in practice it is typically not possible for contaminant to flow between arbitrary a and v_j.

Although not recognized at the time of its introduction, the Berry et al. dynamic SPOP formulation is identical to the well-known p-median facility location problem [Mirchandani and Francis, 1990].[1] In the p-median problem, p actual facilities (e.g., central warehouses) are to be located on m potential sites such that the sum of distances d_{aj} between each of n customers (e.g., retail outlets) a and the nearest facility j is minimized. In contrasting the dynamic SPOP and p-median problems, we observe equivalence between (1) sensors and facilities, (2) attacks and customers, and (3) attack impacts and distances. While Berry et al. allow placement of *at most* p sensors, p-median formulations generally enforce placement of all p facilities; in practice, the distinction is irrelevant unless p approaches the number of possible locations m.

4 Solution Techniques for the p-Median Problem

Equivalence with the p-median problem has an immediate bearing on our approach to solving the dynamic SPOP, as it is now possible to directly leverage the extensive literature on algorithms for solving the p-median problem. The p-median problem, e.g., as defined in §3, can in principle be solved directly as a mixed-integer program (MIP).

[1]We thank Phil Meyers at Pacific Northwest National Laboratory for identifying this relationship.

Further, optimal integer solutions frequently result by relaxing the integral constraints and solving the corresponding pure linear program (LP) [ReVelle and Swain, 1970]. However, due to the rapid growth in the number of constraints and variables as problem size increases, heuristics are often used in practice when dealing with large problem instances. We explore the scalability of LP and MIP approaches to solving dynamic SPOPs in §5.2.

The current state-of-the-art heuristic for the p-median problem is a hybrid approach recently introduced by Resende and Werneck, which we denote *RW*. The core mechanism underlying *RW* is a Greedy Randomized Adaptive Search Procedure (GRASP), which is used to generate a set of high-quality solutions using biased greedy construction techniques. Steepest-descent hill-climbing is then used to move from each of the resulting solutions to a local optimum. Finally, path relinking is used to further explore the set of solutions lying at the intersection of the resulting local optima. For a complete description of *RW*, we refer the reader to [Resende and Werneck, 2004]. We explore the application of *RW* to solving dynamic SPOPs below in §5.3, and contrast the resulting performance with that of the previously described MIP approach.

5 Empirical Results

We now describe the application, performance, and limitations of MIP/LP solvers and the *RW* heuristic for the dynamic SPOP for a number of large-scale, real-world water distribution networks. Our methodology and test networks are detailed in §5.1; results for MIP/LP and heuristic approaches are described in §5.2 and §5.3, respectively.

5.1 Methodology and Test Problems

Our primary objective is to analyze the ability of both MIP solvers and the *RW* heuristic to locate optimal solutions to large-scale instances of the dynamic SPOP. For a given test network, we define the set of attacks A as comprising four distinct possible attacks at each junction, with start times $t_s = 0$, 6, 12, and 18 (units are in hours). Following Berry et al. [2004], each attack consists of a 5500 gallon attack (the storage capacity of a typical water truck) in which contaminant is injected at a rate equal to the outflow rate from the attack vertex v_x. Consequently, the end-time t_f is a function of network hydraulics. EPANET [Rossman, 1999] is used to perform water quality simulations for each attack scenario, and the resulting concentration time-series τ_j are used to compute the impact factors d_{aj} for each combination of $a \in A$ and $v_j \in V$. Simulations begin at time $t_s = 0$ and proceed for a total of 72 hours, i.e., over multiple iterations of the typical demand cycle of 24 hours. As previously indicated, our selection of attack type is arbitrary; the methodology can accommodate any injection scenario supported by EPANET.

We perform empirical studies on a three real-world test networks, which we denote SNL-1, SNL-2, and SNL-3. These networks respectively contain roughly 400, 3000, and 12000 junctions, and 450, 4000, and 14000 pipes. The actual identities, exact dimensions, and pump/valve/tank/reservoir/well counts of these networks are withheld for security purposes. We observe that these models are *not* all-pipes models; the complexity is strictly due to size of the region served by the particular utilities from which the models were obtained. SNL-3 is an order of magnitude larger than any previously considered in the sensor placement optimization literature, and SNL-1 is an order of magnitude larger than that typically investigated. The largest network considered in most analyses, e.g., see [Kessler et al., 1998] and [Ostfeld and Salomons, 2004], is Walski et al.'s "Anytown U.S.A." network [et al., 1987], which consists of 34 pipes, 16 nodes, two tanks, one pump, and one well. Berry et al. [2004] solve a dynamic SPOP via mixed-integer programming for on a network containing roughly 450 junctions and 600 pipes. Watson et al. [2004] examine static SPOP formulations, also in the context of MIP solvers, using both the smaller 450 junction network in addition to a larger network with roughly 3500 junctions.

All experiments are conducted on a dual-processor 64-bit 2.2GHz AMD Opteron workstation with 20 GB of RAM and 60GB of total (RAM plus swap) memory. Despite the "workstation" label, this platform is far more expensive (roughly USD 25K) and powerful than a typical desktop machine, e.g., that found at a typical water utility.

Pre-processing, specifically execution of the water quality simulations, requires non-trivial amounts of computation. For SNL-1, SNL-2, and SNL-3, the respective mean times required to perform water quality analysis for a single attack are approximately 0.75, 1.25, and 4 seconds using EPANET on our workstation. Given four possible attack times per junction, the run-times required to obtain the full suite of water quality simulations range from under an hour for SNL-1 to over 2 days for SNL-3.

5.2 Solution via Mixed-Integer Programming

Much of the prior research on algorithms for both the static and dynamic SPOP involve heuristics, e.g., genetic algorithms. Although many authors claim that their heuristics are capable of locating optimal solutions to test networks, this has never been demonstrated in a rigorous fashion (e.g., via direct comparison with solutions obtained with exact algorithms such as MIP solvers). The only analyses using exact algorithms performed to date have not involved heuristics in any capacity; consequently, no performance bounds on heuristic algorithms are currently available. Berry et al. [2004] solved a compact version of the MIP formulation described in §3 for both (1) a set of attacks at all junctions in a 470-vertex test network and (2) a set of attacks on 100 junctions in a $\approx$ 3,500 vertex test network. Both problems were solved in a matter of hours on a powerful 32-bit workstation with only 4GB of RAM. Given the availability of a more powerful computing platform, we now consider the scalability of MIP formulations of the dynamic SPOP to both larger test networks and test networks with larger sets of attack scenarios.

We use ILOG's[2] AMPL/CPLEX 9.0 MIP solver, which currently represents the state-of-the-art, to compute optimal solutions to the dynamic SPOP for each of our test networks for a range of sensor budgets. The computational results for specific p values, selected to be realistic examples of what might be deployed in practice, are

[2] www.ilog.com.

		Linear Program Statistics			Performance Statistics	
Test Instance	p	Num. Rows	Num. Columns	Num. Non-Zeros	Memory	Run-Time
SNL-1	10	185K	185K	550K	2 GB	26 s.
SNL-2	20	8.5M	8.5M	25M	10 GB	4093 s.
SNL-3	50	27.5M	27.5M	82M	$>$ 30GB	$>$ 3 hrs.

Table 1: Computational results for MIP solution of each of our test networks.

shown in Table 1. All MIPs for SNL-1 and SNL-2 solved without branching, i.e., all variables were integral in the root LP relaxation. As shown in Table 1, the solution times are reasonable, although it is clear that both memory and run-time are a concern. For example, if we consider 24 attack times per junction (one per each hour of a day), then under the best-case assumption of at linear scaling, the LP corresponding to SNL-2 is likely to be intractable. Although we were able to initiate solution of the root LP for SNL-3, the memory requirements are prohibitive and performance was dominated by page swapping; minimal progress was made after 3 hours of computation, at which point we terminated the run.

These experiments identify relatively precise limits on the ability of exact algorithms to solve dynamic SPOPs using modern, high-performance workstations. Specifically, networks with roughly 12,000 junctions and 4 attack times per junction appear to reside at the boundary of what is solvable and what is not. Although not described here, similar boundaries are reached when allowing 24 attack times per junction for 3,000 junction test networks. Although there are clear limits to scalability of the MIP formulation described in §3, we do not view the results presented in this section as negative in any way. When taken in isolation, these results demonstrate the remarkable power of MIP approaches to solving dynamic SPOPs; no other approach has the demonstrated ability to solve test networks as large as SNL-2 under the assumption of multiple attack scenarios per junction. Further, the ability of MIP approaches to identify optimal solutions to large test networks allows us to quantify – in absolute terms – the performance of heuristics for the dynamic SPOP.

5.3 Solution via the *RW* Heuristic

Next, we consider the performance of the *RW* heuristic on each of our test networks; the results are reported in Table 2. On both SNL-1 and SNL-2, *RW* executes in negligible run-times and requires at most modest amounts of memory. Further, the solutions generated by the heuristic are provably *optimal*; the impact I or the total number of gallons of contaminated water consumed is equivalent to that yielded by the exact MIP solvers, as obtained during the course of the experiments described in §5.2. Although not reported, we observe identical behavior on a range of sensor budgets. Relative to the MIP solver, results are obtained 15 to 30 times faster, and require no more than 1/10th of the total memory. However, it is important to note that the heuristic cannot in isolation *prove* the optimality of its result.

On SNL-3, the *RW* heuristic generates a final solution in roughly 29 minutes, while requiring 9 GB of RAM. In contrast, the MIP approach to solving the same test network

			Performance Statistics	
Test Instance	p	Average Consumption	Memory	Run-Time
SNL-1	10	663.8 gallons *	13MB	2 s.
SNL-2	20	2914.3 gallons *	750 MB	130 s.
SNL-3	50	2888.6 gallons	9.0 GB	29 m.

Table 2: Computational results for the *RW* heuristic on each of our test networks. A "*" in the *Average Consumption* column indicates the solution is provably optimal; Average Consumption is defined as the aggregate impact divided by the total number of attacks.

consumed 30 GB of total memory in three hours, eventually failing to find a solution due to excessive swapping. Here, we cannot establish the optimality of the resulting solution; rather, we can only extrapolate behavioral patterns observed on smaller data sets, i.e., we conjecture the resulting solution is optimal. This result illustrates the ability of the heuristic to quickly locate solutions to very large test networks. Further, we observe that this is the largest test network solved to date by *any* algorithmic method; the largest network considered previously involves roughly 3500 junctions, with far fewer attack scenarios.

In our current implementation of *RW*, we use a relatively inefficient database storage scheme, such that 24 of the 29 total minutes required to solve SNL-3 are dedicated to I/O. In preliminary experimentation, we observe that a more compact representation allows us to reduce this time to less than 3 minutes. The memory requirements are significant, in that *RW* cannot currently be executed on a 32-bit platform for networks of this size. As detailed in [Resende and Werneck, 2004], the large memory requirements are due to pre-computations that yield significant run-time improvements. Consequently, it is therefore possible to take the complementary approach and sacrifice run-time for reduced memory requirements.

6 Discussion

At the time we initiated this research, exact solution techniques for both static and dynamic SPOP formulations were reaching limits on 3,000+ junction test networks, due to either the 4GB limit on total memory imposed by 32-bit workstations or excessive computational times exhibited by MIP solvers. Our experiments indicate that while the availability of powerful 64-bit workstations boosts the magnitude of problem we can address via MIP solvers, the increase is not appreciable; Berry et al. [2004] report optimal solutions for 3,500 junction test networks under a limited set of attack scenarios, while we are able to locate optimal solutions to the same test network under attack scenarios at all possible junctions. Larger test networks are not currently soluble by MIP approaches, even when expensive high-performance computing platforms and solvers are available. Consequently, scalability is a major concern, especially given that (1) we expect to encounter real-world problems with at least 50,000 junctions and (2) we would like to consider many more than 4 possible attack times per junction, in order to prevent us from failing to account for rare but high-impact events. Parallel LP solution techniques are one possible avenue to alleviate these issues, and is a route that we are actively pursuing. Finally, despite scalability issues, it is important to note that MIP solvers do play a crucial role in sensor placement optimization, in that they allow us to benchmark heuristic performance in absolute terms. Without such benchmarking, solution quality cannot be assessed – a situation that is unacceptable when deploying systems dedicated to providing maximal public health protection.

Heuristics, and in particular the *RW* algorithm, provide an alternative solution to the scalability problem. Given the extreme difficulty of large-scale test networks for MIP solvers, we fully expected that parallel, high-performance computing would be required to develop effective and scalable heuristics. However, this was not the case; the *RW* algorithm is capable of quickly locating optimal solutions to small-to-medium sized test networks, and can solve 12,000 junction test networks. Further, as discussed in §5.3, the moderate memory requirements of the *RW* algorithm on large networks can be mitigated by an increase in run-time, which is at worst modest. The ability to solve such networks *without* the use of high-performance computing platforms is due to a combination of factors. Most prominently, however, is the fact that the analysis of the mathematical structure of the dynamic SPOP enabled us to recognize the correspondence with the p-median problem and leverage heuristics that efficiently exploit this structure.

It is now clear that high-quality solutions to even very large instances of dynamic SPOP can now be generated using heuristic methods. Although further research is required to resolve specific issues relating to efficiency and scalability, we believe the major focus of future research on the dynamic SPOP should shift from basic algorithmic techniques to exploration of more fundamental engineering issues, including solution robustness [Carr et al., 2004], worst-case optimization objectives, multiple-objective analysis [Watson et al., 2004], and improvement of water quality/transport simulations. Finally, we observe that the major computational bottleneck in solving the dynamic SPOP using heuristic methods is execution of the requisite water quality simulations. Parallelism via execution on a Beowulf cluster is the only currently practical approach that can mitigate the impact of this bottleneck.

7 Conclusions

Researchers have made significant advances in the fidelity of models underlying sensor placement optimization for protection against malicious injection of contaminants in water distribution networks. Any limitations in accuracy are now largely due to the fidelity of the water quality simulations or invalid assumptions relating to the attack scenario, sensor behavior, or emergency response protocols. In contrast, algorithmic advances have lagged the increase in model fidelity. Scalability is a major concern, as all algorithms for high-fidelity models have only been analyzed in the context of relatively small test networks. We have illustrated that exact approaches based on mixed-integer programming can locate optimal solutions to small-to-medium sized test networks, with reasonable computational effort. However, these methods fail to scale to larger test networks. In contrast, state-of-the-art heuristics are capable of locating provably optimal solutions to small-to-medium test networks in significantly shorter run-times than exact approaches, and are able to obtain solutions to very large test networks. This is the first instance in which (1) performance bounds are demonstrated for heuristic methods for sensor placement and (2) scalability of a heuristic method is conclusively demonstrated. Consequently, our results serve as a yardstick for future research on algorithms for sensor placement. In particular, we emphasize the necessity for moving beyond "toy"-sized test networks and, in the case of heuristics, demonstrating performance relative to known optimal solutions.

References

J. Berry, L. Fleischer, W.E. Hart, and C.A. Phillips. Sensor placement in municipal water networks. In *Proceedings of the World Water and Environmental Resources Conference*, 2003.

J. Berry, W.E. Hart, C.A. Phillips, and J. Uber. A general integer-programming-based framework for sensor placement in municipal water networks. In *Proceedings of the World Water and Environmental Resources Conference*, 2004.

R. Carr, H. J. Greenberg, W. E. Hart, and C. A. Phillips. Addressing modeling uncertainties in sensor placement for community water systems. In *Proceedings of the World Water and Environmental Resources Conference*, 2004.

T. Walski et al. Battle of the network models: Epilogue. *Journal of Water Resources Planning and Management*, 113(2):191–203, 1987.

A. Kessler, A. Ostfeld, and G. Sinai. Detecting accidental contaminations in municipal water networks. *Journal of Water Resources Planning and Management*, 124(4): 192–198, July/August 1998.

P.S. Mirchandani and R.L. Francis, editors. *Discrete Location Theory*. John Wiley and Sons, 1990.

A. Ostfeld and E. Salomons. Optimal layout of early warning detection stations for water distribution systems security. *Journal of Water Resources Planning and Management*, 130(5):377–385, 2004.

M.G.C. Resende and R.F. Werneck. A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10(1):59–88, 2004.

C. ReVelle and R. Swain. Central facilities location. *Geographical Analysis*, 2:30–42, 1970.

L.A. Rossman. The EPANET programmer's toolkit for analysis of water distribution systems. In *Proc. of the Annual Water Resources Planning and Management Conference*, 1999.

J.P. Watson, H.J. Greenberg, and W.E. Hart. A multiple-objective analysis of sensor placement optimization in water networks. In *Proceedings of the World Water and Environmental Resources Conference*. ASCE, 2004.